**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

**отчет**

по лабораторной работе №**3**

по дисциплине «Операционные системы»

на тему: Исследование организации управления основной памятью

|  |  |  |
| --- | --- | --- |
| Студент гр. 6381 |  | Кухарев М.А. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

**Цель работы:**

Для исследования организации управления памятью необходимо ориентироваться на тип основной памяти, реализованный в компьютере и способ организации, принятый в ОС. В лабораторной работе рассматривается нестраничная память и способ управления динамическими разделами. Для реализации управления памятью в этом случае строится список занятых и свободных участков памяти. Функции ядра, обеспечивающие управление основной памятью, просматривают и преобразуют этот список.

В лабораторной работе исследуются структуры данных и работа функций управления памятью ядра операционной системы.

**Описание функций:**

|  |  |
| --- | --- |
| GET\_AVAILABLE\_MEMORY | Получение размера доступной памяти |
| GET\_EXTENDED\_MEMORY | Получение размера расширенной памяти |
| GET\_MCB\_DATA | Получение данных одного MSB блока |
| GET\_ALL\_MSB\_DATA | Получение данных со всех MSB блоков |
| PRINT\_STRING | Вывод строки на экран |
| WRD\_TO\_HEX | Переводит число AX в строку в 16-ой с/с, записывая получившееся в di, начиная с младшей цифры |
| BYTE\_TO\_DEC | Переводит байт из AL в десятичную с/с и записывает получившееся число по адресу si, начиная с младшей цифры |
| BYTE\_TO\_HEX | Переводит число AL в коды символов 16-ой с/с, записывая получившееся в al и ah |
| TETR\_TO\_HEX | Вспомогательная функция для работы функции BYTE\_TO\_HEX. |

**Необходимые сведения и описание структур данных:**

Учет занятой и свободной памяти ведется при помощи списка блоков управления памятью MCB (Memory Control Block). MCB занимает 16 байт (параграф) и располагается всегда с адреса кратного 16 (адрес сегмента ОП) и находится в адресном пространстве непосредственно перед тем участком памяти, которым он управляет.

MCB имеет следующую структуру:

|  |  |  |
| --- | --- | --- |
| Смещение | Длина поля (байт) | Содержимое поля |
| 00h | 1 | тип MCB:  5Ah, если последний в списке,  4Dh, если не последний |
| 01h | 2 | Сегментный адрес PSP владельца участка памяти, либо  0000h - свободный участок,  0006h - участок принадлежит драйверу  OS XMS UMB  0007h - участок является исключенной верхней памятью драйверов  0008h - участок принадлежит MS DOS  FFFAh - участок занят управляющим блоком 386MAX UMB  FFFDh - участок заблокирован 386MAX  FFFEh - участок принадлежит 386MAX UMB |
| 03h | 2 | Размер участка в параграфах |
| 05h | 3 | Зарезервирован |
| 08h | 8 | "SC" - если участок принадлежит MS DOS, то в нем системный код  "SD" - если участок принадлежит MS DOS, то в нем системные данные |

По сегментному адресу и размеру участка памяти, контролируемого этим MCB можно определить местоположение следующего MCB в списке.

Адрес первого MCB хранится во внутренней структуре MS DOS, называемой "List of Lists" (список списков). Доступ к указателю на эту структуру можно получить, используя функцию f52h "Get List of Lists" int 21h. В результате выполнения этой функции ES:BX будет указывать на список списков. Слово по адресу ES:[BX-2] и есть адрес самого первого MCB.

Размер расширенной памяти находится в ячейках 30h, 31h CMOS. CMOS это энергонезависимая память, в которой хранится информация о конфигурации ПЭВМ. Объем памяти составляет 64 байта. Размер расширенной памяти в Кбайтах можно определить обращаясь к ячейкам CMOS следующим образом:

mov AL,30h ; запись адреса ячейки CMOS

out 70h,AL

in AL,71h ; чтение младшего байта

mov BL,AL ; размера расширенной памяти

mov AL,31h ; запись адреса ячейки CMOS

out 70h,AL

in AL,71h ; чтение старшего байта

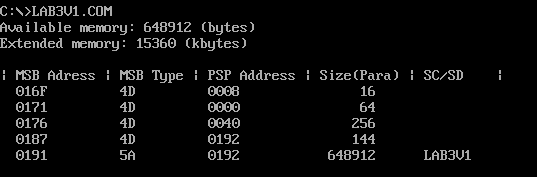
; размера расширенной памяти

**Ход работы:**

Результат работы показан на скриншоте ниже.

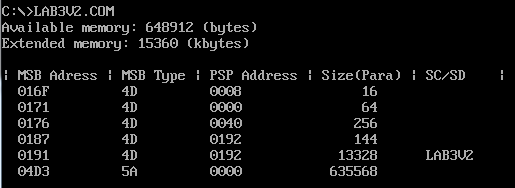
**Pre.S. Размер участка (Size (Para)) указан в байтах.**

1. LAB3V1.COM



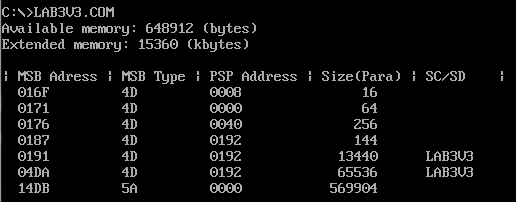
Из рисунка видно, что программа занимает максимум памяти, потому что при запросе размера доступной памяти мы выделяем, по сути, столько памяти, сколько возможно.

2. LAB3V2.COM



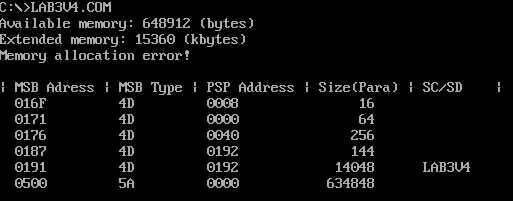
В данном случае мы освобождаем память. В итоге остается столько памяти, сколько занимает программа. После освобождения памяти, как видно на рисунке, есть блок свободной памяти, из которого, если вдруг нам потребуется ещё, будет выделятся память.

3. LAB3V3.COM



В данном случае мы сначала выделяем всю доступную память, потом освобождаем то, что не нужно. Затем запрашиваем блок памяти 64 кб, в итоге система выделяет нам ещё 64 кб памяти.

4. LAB3V4.COM



В данном случае мы выделяем всё доступную память, а затем ещё запрашиваем 64 кб. В результате возникает ошибка. Она возникает из-за того, что мы в первый раз уже выделили всё доступную память, т.е. больше выделить уже нельзя, но мы всё равно пытаемся, и в итоге получаем ошибку.

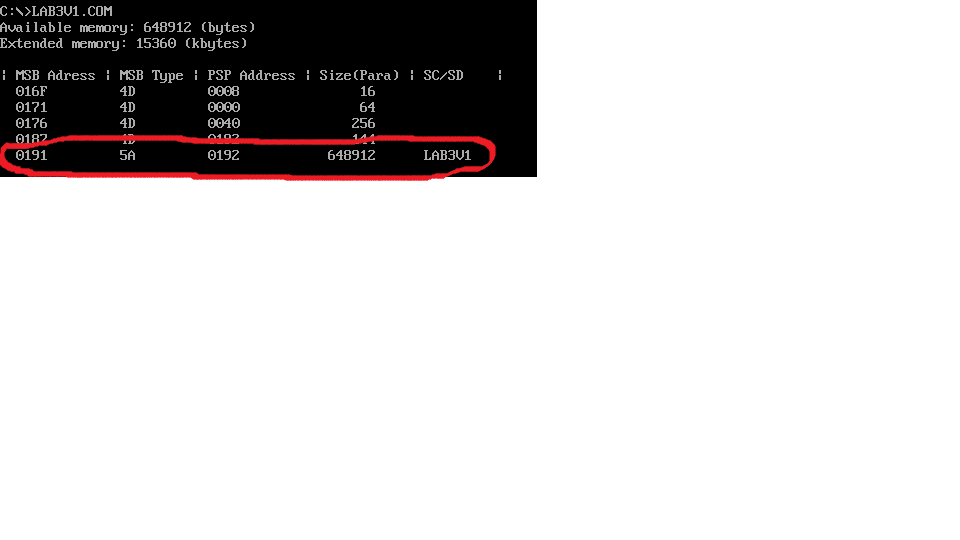
**Контрольные вопросы:**

1. Что означает «доступный объём памяти»?

Ответ: Это максимальный объем памяти, который может быть доступен программе. В этом мы убеждаемся в четвёртом пункте данной л.р., когда после выделения всей доступной памяти, мы пытаемся выделить ещё, но, увы, – нельзя.

1. Где MCB блок Вашей программы в списке?

Ответ: Блок нашей программы выделен красным на рисунке.



1. Какой размер памяти занимает программа в каждом случае?

Ответ:

LAB3V1.com 648912 байт.

LAB3V2.com 13328 байт.

LAB3V3.com 13440 байт (без блока в 64кб).

LAB3V4.com 14048 байт.